

**LAPORAN PRAKTIKUM**  
**MODUL 4**  
**“SINGLE LINKED LIST (BAGIAN PERTAMA)”**



**Disusun oleh:**

**Wendri Tri Pambudi (21104048 / SE07-01)**

**Dosen:**

**Yudha Islami Sulistya, S.Kom., M.Cs**

**PROGRAM STUDI REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**UNIVERSITAS TELKOM PURWOKERTO**

**2024**

## Tugas Pendahuluan

### 1. Membuat deklarasi tipe List

Buat file list.h

```
04_Single_Linked_List_Bagian_1 > TP > h list.h > address
1  #include <iostream>
2
3  #define first(L) (L).first
4  #define next(P) (P)->next
5  #define info(P) (P)->info
6
7  using namespace std;
8  typedef int infotype;
9  typedef struct elmList *address;
10
11  struct elmList {
12      infotype info;
13      address next;
14  };
15
16  struct List{
17      address first;
18  };
19
```

Buat file list.cpp

```
04_Single_Linked_List_Bagian_1 > TP > C++ list.cpp > allocate(infotype)
1  #include <iostream>
2  #include "list.h"
3  using namespace std;
4
```

### 2. Membuat list kosong, yaitu procedure createList.

Tambahkan pada list.h primitif dari procedure createList

```
void createList(List &L);
```

Implementasi dari procedure createList

```
void createList (List &L) {
    first(L)= NULL;
}
```

### 3. Setelah list sudah ada, selanjutnya buatlah elemen dengan menggunakan fungsi allocate.

Tambahkan pada list.h primitif dari fungsi allocate

```
address allocate(infotype x);
```

Tambahkan pada list.cpp implementasi dari fungsi allocate

```
address allocate(infotype x){  
    address p = new elmList;  
    info(p) = x;  
    next(p) = NULL;  
  
    return p;  
}
```

4. Setelah List dan elemen sudah ada, maka selanjutnya elemen tersebut harus diinsert ke List agar bisa menjadi elemen list.

Tambahkan pada list.h primitif procedure insertFirst

```
void insertFirst(List &L, address P);
```

Tambahkan pada list.cpp implementasi dari procedure insertFirst

```
void insertFirst(List &L, address P){  
    next(P) = first(L);  
    first(L) = P;  
}
```

5. Setelah proses insert elemen, maka agar bisa mengetahui apakah elemen berhasil diinsertkan, maka kita perlu menampilkan isi list.

Tambahkan pada list.h primitif procedure printInfo

```
void printInfo(List L);
```

Tambahkan pada list.cpp implementasi dari proc printInfo

```
void printInfo (List L){  
    address p = first(L);  
    while (p != NULL){  
        cout << info(p) << ", ";  
        p = next(p);  
    }  
    cout << endl;  
}
```

6. Sekarang, setelah ADT List sudah terisi dengan beberapa fungsi Procedur di atas, maka mari buat sebuah List berisi 3 elemen yang berisi 3 digit nim terakhir Anda di main.cpp

Code main.cpp:

```
int main() {
    List L;
    createList(L); // 1. Panggil createlist

    int input; // Variabel untuk input angka dari user
    address p;

    // 2. Menanyakan angka pertama dari user
    cout << "Masukkan angka pertama: ";
    cin >> input;
    // 3. Panggil fungsi allocate agar data tersebut disajikan elemen
    p = allocate(input);
    // 4. Panggil prosedur insertFirst untuk memasukkan elemen ke list
    insertFirst(L, p);
    // 5. Panggil prosedur printInfo untuk menampilkan isi list
    printInfo(L);

    // 6. Menanyakan angka kedua dari user
    cout << "Masukkan angka kedua: ";
    cin >> input;
    p = allocate(input);
    insertFirst(L, p);
    printInfo(L);

    // 7. Menanyakan angka ketiga dari user
    cout << "Masukkan angka ketiga: ";
    cin >> input;
    p = allocate(input);
    insertFirst(L, p);
    printInfo(L);

    return 0;
}
```

Output:

```
Masukkan angka pertama: 0
0,
Masukkan angka kedua: 4
4, 0,
Masukkan angka ketiga: 8
8, 4, 0,
```

## 7. SESI HAVE FUN

1. Tambahkan procedure insertLast, insertAfter, deleteLast, deleteAfter pada list.h dan list.cpp
2. Tambahkan Function searchInfo pada list.h dan list.cpp
3. Ubah main.cpp agar proses insert N data tidak satu persatu, tapi sesuai dengan jumlah digit NIM yaitu 10 data (clue : gunakan looping). Dan NIM

yang diinput, saat di show tidak boleh terurut terbalik (clue : gunakan insert Last) Tampilan

Jawaban:

1. Code procedure tambahan insertLast, insertAfter, deleteLast, deleteAfter pada list.h dan list.cpp :

Code list.cpp

```
04_Single_Linked_List_Bagian_1 > TP > C++ list.cpp > insertAfter(address, address)
44 //tambahan procedure
45 void insertAfter(address Prec, address P) {
46     if (Prec != NULL) {
47         next(P) = next(Prec);
48         next(Prec) = P;
49     }
50 }
51
52 void deleteLast(List &L, address &P) {
53     if (first(L) == NULL) {
54         P = NULL;
55     } else if (next(first(L)) == NULL) {
56         P = first(L);
57         first(L) = NULL;
58     } else {
59         address last = first(L);
60         address prevLast = NULL;
61         while (next(last) != NULL) {
62             prevLast = last;
63             last = next(last);
64         }
65         P = last;
66         next(prevLast) = NULL;
67     }
68 }
```

```
04_Single_Linked_List_Bagian_1 > TP > C++ list.cpp > insertAfter(address, address)
52 void deleteLast(List &L, address &P) {
69
70 void deleteAfter(address Prec, address &P) {
71     if (Prec != NULL && next(Prec) != NULL) {
72         P = next(Prec);
73         next(Prec) = next(next(Prec));
74     }
75 }
76
77
```

code list.h

```
04_Single_Linked_List_Bagian_1 > TP > h list.h > ...
24
25 // Prosedur tambahan
26 void insertLast(List &L, address P);
27 void insertAfter(address Prec, address P);
28 void deleteLast(List &L, address &P);
29 void deleteAfter(address Prec, address &P);
```

2. Tambahkan Function searchInfo pada list.h dan list.cpp

Code list.cpp

```
04_Single_Linked_List_Bagian_1 > TP > C++ list.cpp > ...
77     address searchInfo(List L, infotype x) {
78         address p = first(L);
79         while (p != NULL) {
80             if (info(p) == x) {
81                 return p;
82             }
83             p = next(p);
84         }
85         return NULL;
86     }
87
```

Code list.h

```
04_Single_Linked_List_Bagian_1 > TP > h list.h > searchInfo(List, infotype
30
31     address searchInfo(List L, infotype x);
```

3. Ubah main.cpp agar proses insert N data tidak satu persatu, tapi sesuai dengan jumlah digit NIM yaitu 10 data

Code main.cpp

```
04_Single_Linked_List_Bagian_1 > TP > C++ list.cpp > main()
123     //SESI HAVE FUN
124     int main() {
125         List L;
126         createList(L);
127
128         int digit;
129         address p;
130         cout << "Masukkan NIM perdigit\n";
131
132         for (int i = 1; i <= 10; i++) {
133             cout << "Digit " << i << ": ";
134             cin >> digit;
135
136             p = allocate(digit);
137
138             insertLast(L, p);
139         }
140
141         cout << "Isi list : ";
142         printInfo(L);
143
144         return 0;
145     }
```

Output:

```
Masukkan NIM perdigit
Digit 1: 0
Digit 2: 0
Digit 3: 2
Digit 4: 1
Digit 5: 1
Digit 6: 0
Digit 7: 4
Digit 8: 0
Digit 9: 4
Digit 10: 8
Isi list : 0, 0, 2, 1, 1, 0, 4, 0, 4, 8,
```